# Algorithmic Trading Bot Using Machine Learning

*Submitted by*

**Gaurang Nayee**

*In Partial Fulfillment of the Requirements for the Degree of*

**Master of Science (M.Sc.)**


**Indian Institute of Technology Jodhpur**

**Department of Mathematics**

*December, 2024*

# Declaration

This is to affirm that the work presented in this Project Report titled *Algorithmic Trading Bot Using Machine Learning*, submitted to the Indian Institute of Technology Jodhpur as part of the requirements for the degree of M.Sc., is an original record of the research conducted under the guidance of Dr. Vivek Vijay and Dr. Sandeep Kumar Yadav. To the best of my knowledge, the content of this report, in its entirety or in part, has not been submitted, nor will it be submitted, to any other institution or university, either in India or abroad, for the purpose of obtaining any degree or diploma.

**Signature**

*Gaurang Nayee*

M23MA1007

# Certificate

This is to certify that the Project Report titled *Algorithmic Trading bot using Machine Learing*, submitted by Gaurang Nayee (M23MA1007) to the Indian Institute of Technology Jodhpur in partial fulfillment of the requirements for the degree of M.Sc. Mathematics, is an authentic record of the research work carried out by him under my supervision. To the best of my knowledge, the content of this report, either in whole or in part, has not been submitted to any other institution or university for the award of any degree or diploma.

**Signature**

Dr.Sandeep Kumar Yadav & Dr.Vivek Vijay

# Acknowledgements

I would like to express my deepest gratitude to my project supervisor, Dr.Sandeep Kumar Yadav & Dr.Vivek Vijay , for their invaluable guidance, support, and encouragement throughout the course of this project. Their insightful feedback and expertise have been instrumental in shaping this work.

I am also thankful to the Department of Mathematics at the Indian Institute of Technology Jodhpur for providing me with an excellent academic environment and the necessary resources to complete this project.

My sincere thanks go to my peers and friends, who offered constructive discussions and moral support, making this journey a collaborative and enriching experience.

Finally, I would like to extend my heartfelt appreciation to my family for their unwavering support and encouragement, which have been my foundation during the entire course of my M.Sc. studies.

This project, *Algorithmic Trading Bot Using Machine Learning*, has been a significant milestone in my academic journey, and I am grateful to all who contributed to its success.

# Abstract

The technical indicators are highly uncertain, therefore possess greater influence on stock market prediction. Among different techniques developed for effective prediction of the financial market, the AI techniques show better prediction efficiency. In this paper, a hybrid model combined with autoencoder (AE) and kernel extreme learning machine (KELM) is proposed for further improvement in the quality of financial market prediction. This study mainly emphasizes on a precise prediction of the financial market. The main motive behind stock price prediction is minimizing the substantial losses faced by investors and analyzing the profitability with the help of buying and selling amount. The prime advantage of the proposed technique over the conventional SAE is robust prediction of different financial markets with reduction in error. To authenticate the performance of the proposed deep learning (DL) technique (KELM-AE), high-frequency data of different financial markets like Yes Bank, SBI, ASHR, and DJI are taken into consideration, and the performance of the proposed technique is investigated in Python-based simulation in accordance with MAPE (Mean Absolute Percentage Error), MAE (Mean Absolute Error), and RMSE (Root Mean Square Error). The application of SAE is new in the field of predicting different bank data. The validation of the model is performed by comparing it with other traditional methods based on different performance indexes.

# Contents

# List of Figures



# List of Tables

# Algorithmic Trading bot using Machine Learing

## 1 Introduction

The prediction of financial markets is essential for minimizing investor risks. This work utilizes the Kernel Extreme Learning Machine (KELM) model combined with deep learning approaches such as Stacked Autoencoders (SAE) to enhance prediction accuracy. The model aims to reduce errors in forecasting volatile markets, including major institutions like Bank of India, SBI, YES Bank.

## 2 Problem Definition and Objective

### 2.1 Problem Definition

The financial market is inherently volatile and complex, making accurate prediction of stock prices a challenging task. Traditional methods often struggle to capture the intricate patterns and nonlinear relationships present in financial data. This limitation leads to suboptimal investment decisions and increased risks for traders and investors.

The key challenges include:

- High volatility and noise in financial time-series data.
- Difficulty in extracting meaningful features from large datasets.
- Limitations of traditional machine learning models in handling nonlinear relationships.
- Overfitting issues in predictive models, reducing their reliability in real-world scenarios.

### 2.2 Objective

The primary objective of this project is to develop an accurate and efficient algorithmic trading bot that leverages advanced machine learning techniques to predict financial market trends. Specifically, the focus is on combining Stacked Autoencoders (SAE) and Kernel Extreme Learning Machine (KELM) to:

- Extract relevant features from financial data to improve predictive accuracy.
- Develop a robust model capable of handling nonlinear relationships and reducing overfitting.
- Generate actionable Buy/Sell signals based on predicted prices to assist traders in making informed decisions.
- Evaluate the model's performance using metrics such as MAPE, MAE, and RMSE.
- Test the strategy returns based on the generated trading signals to demonstrate the profitability of the approach.

The ultimate goal is to create a system that minimizes prediction errors while maximizing trading profits, providing a reliable solution for financial market prediction and decision-making.

# 3 Methodology

## 3.1 Kernel Extreme Learning Machine (KELM)

KELM enhances traditional ELM by incorporating kernel functions, enabling better generalization. The output is represented as:

$$f(x) = \sum_{i=1}^{L} \beta_i h_i(x)$$

where $\beta$ is the weight vector and $h(x)$ is the feature mapping function. Various kernels, such as polynomial, Gaussian, and wavelet, are used to map data to higher-dimensional spaces.

# 4 Algorithm

The proposed algorithm combines feature extraction using Stacked Autoencoders (SAE) with prediction capabilities of the Kernel Extreme Learning Machine (KELM). The steps are outlined below:

1. **Input:** Financial data, such as Open, High, Low, and Close (OHLC) values.
2. **Step 1:** Normalize the input financial data to ensure uniformity and scalability.

```
from sklearn.preprocessing import MinMaxScaler

# Scaling the data
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
```

3. **Step 2:** Train a Stacked Autoencoder (SAE) to extract meaningful features from the normalized data.

```
from tensorflow.keras import layers, models

# Define SAE
input_layer = layers.Input(shape=(X_train.shape[1],))
encoded = layers.Dense(32, activation='relu')(input_layer)
decoded = layers.Dense(X_train.shape[1], activation='sigmoid')(encoded)

autoencoder = models.Model(input_layer, decoded)
```

```
autoencoder.compile(optimizer='adam', loss='mse')
autoencoder.fit(X_train, X_train, epochs=50, batch_size=32, validation_data=(X_test, X_test)

# Encode features
encoder = models.Model(input_layer, encoded)
X_train_encoded = encoder.predict(X_train)
X_test_encoded = encoder.predict(X_test)
```

4. **Step 3:** Use the extracted features as input to the KELM for prediction.

```
# Placeholder for KELM implementation with encoded features
# Use extracted X_train_encoded and X_test_encoded for predictions
```

5. **Step 4:** Optimize the output weights of KELM using the following formula:

$$\beta = (I/C + H^T H)^{-1} H^T T, \quad where C is the regularization parameter.$$

6. **Step 5:** Compute error metrics, such as:

   - Mean Absolute Percentage Error (MAPE)
   - Mean Absolute Error (MAE)
   - Root Mean Square Error (RMSE)

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Calculate errors
mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print("MAE:", mae)
print("RMSE:", rmse)
```

7. **Step 6:** Generate actionable Buy/Sell signals based on predictions:

```
buy_signals = y_pred > y_test
sell_signals = y_pred < y_test
```

8. **Output:** Predicted financial data values (e.g., closing price) and trading signals.

The flow of the algorithm is depicted in Figure 4.1.

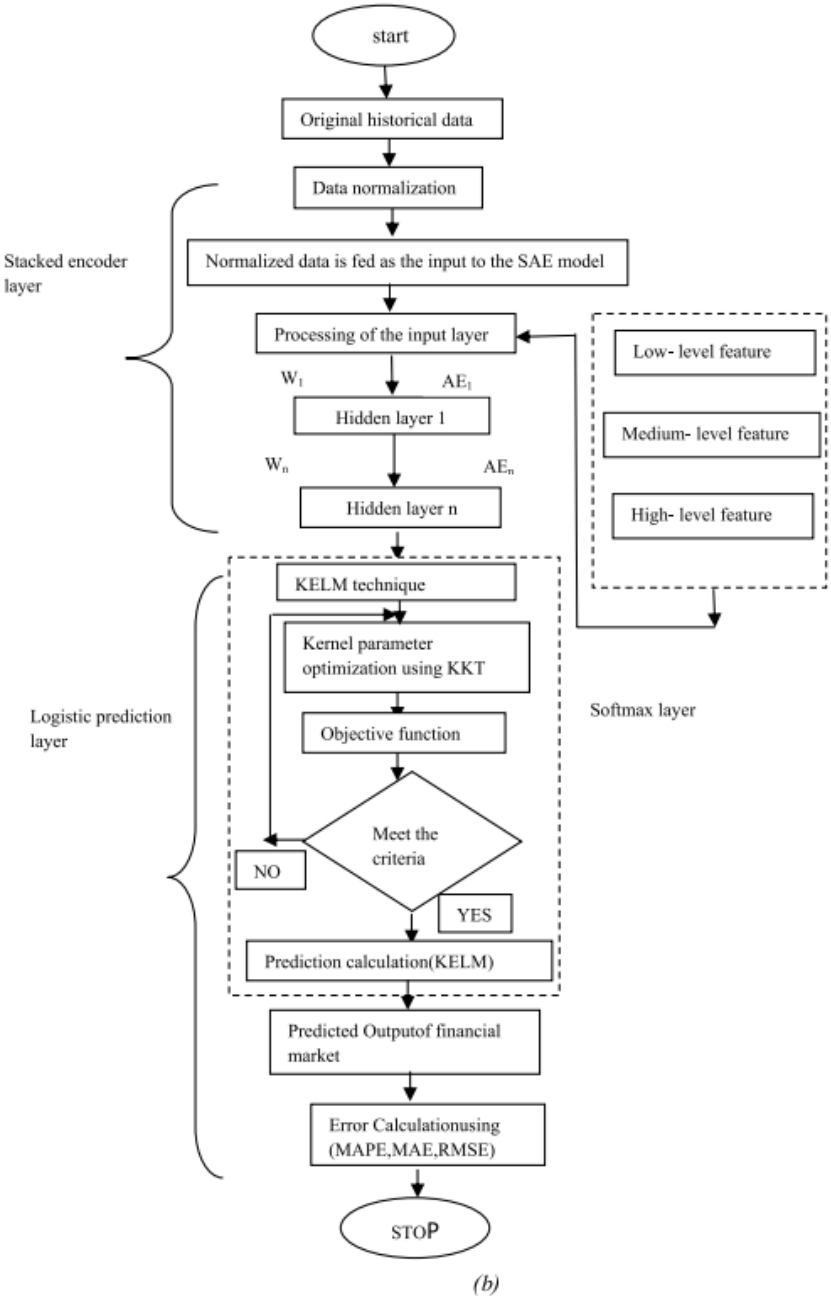


Figure 4.1: Flow of the Algorithm

# 5 Performance Evaluation

Metrics such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) were used to evaluate the model. The proposed KELM-AE model showed improved prediction accuracy and generated actionable Buy/Sell signals.

The performance metrics for different banks are summarized in the table below:

| Bank | MAPE (%) | MAE | RMSE | R-squared |
|---|---|---|---|---|
| **BOI** | 17.95% | 7.1318 | 8.5811 | 0.7288 |
| **SBI** | 0.50% | 0.9570 | 1.6366 | 0.9994 |
| **Yes Bank** | 9.85% | 3.2493 | 3.5730 | 0.9364 |

Table 5.1: Performance Metrics for BOI, SBI, and Yes Bank

## 5.1 Predicted vs Actual Prices

The following figures show the comparison between the predicted and actual closing prices for the entire dataset from 2013-11-06 to 2020-06-12 for each bank.

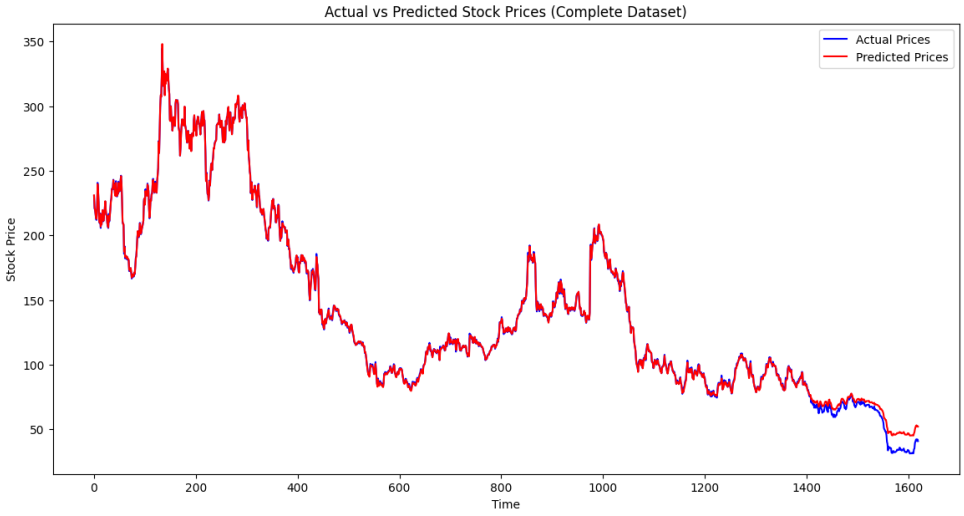


Figure 5.1: Predicted vs Actual Closing Prices for Bank of India (BOI)

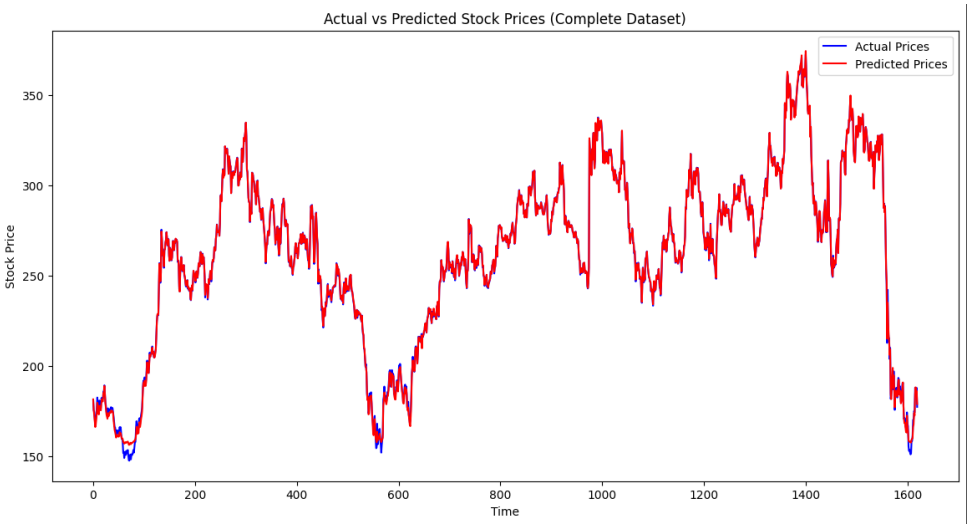


Figure 5.2: Predicted vs Actual Closing Prices for State Bank of India (SBI)

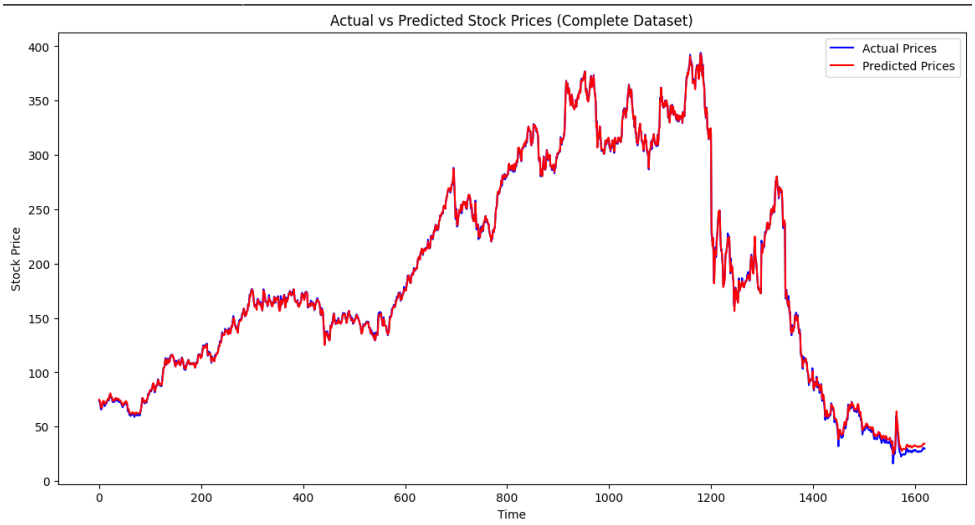


Figure 5.3: Predicted vs Actual Closing Prices for Yes Bank

# 6 Summary and Future plan of work

The proposed KELM-AE model demonstrates superior performance compared to traditional methods, delivering improved prediction accuracy with minimal error. Building upon this success, the next step involves fine-tuning the model and deploying it using an API key for live trading in a paper trading environment. This deployment will allow for testing the model's performance in real-world scenarios while ensuring the risk-free assessment of its predictions. — Additionally, the focus will be on evaluating the model's effectiveness for various types of shares, particularly examining its performance on more volatile stocks. Insights into the model's limitations and the underlying causes of any inaccuracies will guide further enhancements. Addressing these challenges and identifying the factors affecting its performance will be pivotal in refining the model to achieve better results and broader applicability in financial market predictions.

# 7 References

The methodology used in this project is heavily influenced by the work of Mohanty et al. [**?**], who introduced the combination of Autoencoder and Kernel Extreme Learning Machine for financial market prediction. This framework has been proven to enhance prediction accuracy and reduce overfitting, making it suitable for volatile financial data.